IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of: Confirmation No.: 7312

Bhaskar Ghosh Examiner: Hwa, S.

Serial No.: 10/824,887 Group Art Unit No.: 2169

Filed on: April 13, 2004

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

This Appeal Brief is submitted in support of the Notice of Appeal filed on October 31, 2007. A Panel Decision from Pre-Appeal Brief Review was mailed December 27, 2007. The time period for filing this Appeal Brief is extended to January 28, 2008, as January 27, 2008, the one month date from the mailing of the Panel Decision, is a Sunday.

I. REAL PARTY IN INTEREST

Oracle International Corporation is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

The instant application shares a common parentage with Application No. 10/944,975 (Attorney Docket No. 502777-2403) filed on September 16,2004, which is also presently under Appeal. Both applications claim benefit of priority from U.S. Provisional Application No. 60/530,413, filed December 16, 2003. This related Appeal is also docketed within Art

Unit 2169, Examiner S. Hwa. Appellants are unaware of any other related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-25 and 27-29 have been finally rejected and are the subjects of this appeal.

IV. STATUS OF AMENDMENTS

The claims were not amended after the Final Office Action.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present application contains independent Claim 1, which is summarized below. Claims 2, 6, 7, and 11 are argued separately from the independent Claim 1 upon which they depend, and therefore are also summarized below.

The claims summarized below are annotated to cross-reference features of the claims to specific examples of those features disclosed in the specification. However, the annotations are not intended to limit the scope of the recited features to those specific examples to which the annotations refer.

Claim 1 recites (with reference annotations in parenthesis) a method for processing a database statement within a database server, the method comprising the steps of: receiving at the database server the database statement (par. [0026]); determining that at least one operation required by the database statement can be parallelized (par. [0027]); within the database server, generating a set of information about how to execute the database statement; causing a plurality of slave processes to perform said at least one operation by sharing the set of information with each (par. [0028]) slave process of said plurality of slave processes,

wherein the set of information shared with each slave process includes (a) information about a task to be performed by said slave process, and (b) information about one or more tasks, to be performed by processes other than the slave process (par. [0028]), to execute the database statement (par. [0029] and [0068]); and sending to each slave process of said plurality of slave processes data (par. [0038]) that indicates which part (par. [0029]) of the set of information shared with the slave process represents the part of the at least one operation that should be performed by the slave process.

Claim 2 recites (with added reference annotations in parenthesis) a method similar to that of Claim 1, wherein the step of sharing the set of information includes sharing an execution plan (par. [0030]) for the database statement; and sharing the execution plan with a particular slave process of the plurality of slave processes is performed by: providing an original statement of the database statement to a node (par. [0064]) on which the particular slave process resides, wherein the original statement is the form of the database statement in which the database statement was received by the database server; at said node, generating an equivalent execution plan (par. [0068]) based on the original statement; and the particular slave process accessing the equivalent execution plan.

Claim 6 recites (with added reference annotations in parenthesis) a method similar to that of Claim 1 wherein the step of generating a set of information includes generating an execution plan for the database statement; constructing a shared cursor for the database statement, wherein the shared cursor provides access to the execution plan; and the step of sharing access includes providing each slave process of said plurality of slave processes access to the shared cursor.

Claim 7 recites (with added reference annotations in parenthesis) a method similar to that of Claim 6 wherein the step of providing each slave process of said plurality of slave processes access to the shared cursor includes allowing two or more of said slave processes to access a shared instance of the shared cursor.

Claim 11 recites (with added reference annotations in parenthesis) a method similar to that of Claim 1 wherein the step of generating a set of information includes generating an execution plan for the database statement, wherein the set of information includes the execution plan; and the method further comprises the step of inserting into the execution plan a granule iterator row source that encapsulates a horizontal partitioning of a base object upon which the database statement operates.

Claims 15-25 and **27-29** recite computer readable storage media (FIG. 3, storage device 310) that carry instructions for causing processors (FIG. 3, processor 304) to perform the steps of the methods of Claims 1-14, respectively.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-3 and 15-17 stand rejected as allegedly anticipated under 35 U.S.C. § 102(e) by Ratcliff, U.S. Patent No. 6,996,596 ("RATCLIFF").

2. Claims 2-8 and 12-18 stand rejected as allegedly unpatentable under 35 U.S.C. § 103(a) over RATCLIFF in view of Hallmark, U.S. Patent No. 5,857,180 ("HALLMARK").

VII. ARGUMENTS

A. The Features of Claims 1 and 15 Are Not Disclosed, Taught, or Suggested by Ratcliff

Claim 1 recites one or more features that are not taught or suggested by Ratcliff. For example, as highlighted above, Claim 1 recites, inter alia, processing a database statement within a database server. Conversely, Ratcliff not directed toward databases, database servers, or database statements. Instead Ratcliff is directed to a plurality of processors cooperating to solve a job such as an algorithm (Ratcliff, col. 3, lines 28-30).

The Response to Applicant's Arguments within the Final Office Action mailed August 31, 2007 asserts that Ratcliff teaches a first column of a table includes a list of numbers or other information that identify various processors. This section goes on to state that "[i]n a larger scale example, this table is in the form of a master database accessible by the originating server

(Final Office Action, Page 2, Section 3, citing Ratcliff at column 5, lines 62 to column 6, line 2).

These remarks misquote/mischaracterize Ratcliff, and do not address processing a database statement within a database server, as claimed. Ratcliff's "list of algorithms and/or data delivered to the respective processors" does not correspond with the claimed database statements. Anticipation under 35 U.S.C. §102 is established only when a single prior art reference discloses, expressly or under principles of inherency, each and every element of a claimed invention. RCA Corp. v. Applied Digital Data Systems, Inc., 730 F.2d 1440, 1444, 221 U.S.P.Q. (BNA) 385, 388 (Fed. Cir.). For at least the above reasons, the rejection of claim 1 is invalid and should be withdrawn.

The Response to Applicant's Arguments goes on to state that Ratcliff discloses a software platform upon which application programs (e.g. include any SQL statement) may execute (Final Office Action, Page 3, Paragraph 1). Applicant notes that the above parenthetical "(e.g. include any SQL statement)" was inserted by the Examiner, and is not contained within the original Ratcliff document. This is significant because Applicant is explicitly claiming "processing a database statement within a database server", and the Office Action is attempting to stretch application programs to include any SQL statement, even though this feature is not disclosed by Ratcliff either expressly or under principles or inherency. Thus, the rejection of Claim 1 is defective and should be withdrawn.

Ratcliff also does not disclose generating a set of information about how to execute a database statement, as claimed. The portions of Ratcliff relied upon by the Final Office Action ("packaging" data, sending over network as an object; col. 4, lines 56-63) are mis-applied. The Final Office Action is correct that Ratliff's data is packaged and sent. However, nowhere

within the cited section, or any part of Ratcliff, is the cited data described as containing any information about how to execute anything, much less a database statement. Additionally, if no database statements are disclosed, it is not possible to anticipate generating a set of information about how to execute a database statement.

The Response to Applicant's Arguments addresses this by stating that Ratcliff teaches that the instruction are executed by the process of the one data processing device to perform the portion of the algorithm on the data. The reward is provided to the recipient associated with the one data processing device (Final Office Action, Page 3, Paragraph 3, quoting/paraphrasing Ratcliff's column 2, lines 25-29).

The key to understanding this argument is in the word "portion". Applicant agrees that any one of Ratcliff's processors may operate on a portion of data. However, Ratcliff is silent as to how this occurs. Apparently, the Examiner intends for Ratcliff's algorithm corresponds to claimed "set of information about how to execute", but the reasoning behind this (purported) correspondence is not stated, nor does the correspondence in fact exist.

Ratcliff further does not contain slave processes. The Office Action cites Ratcliff's Distributed Objects Environment (DOE) as anticipating the claimed slave processes (Office Action, page 4, paragraph 5; citing Ratcliff, col. 5, lines 15-20). However, this characterization is erroneous. Ratcliff does not describe any type of distribution of information across each DOE, and thus cannot correspond to the claimed "sharing the set of information with each slave process of said plurality of slave processes".

The Response to Applicant's Arguments addresses this by stating that Ratcliff teaches the data processing system include a plurality of processors (e.g. slaves, elements 104-130 of figure 2 and 3; see also, server, element 210) (Final Office Action, Page 3, Paragraphs 5 and 6).

However, this citation is inadequate because it is not responsive to Applicant's arguments, and does not make any explanation regarding the deficiency of the rejection. As stated in RCA Corp., anticipation under 35 U.S.C. §102 is established only when a single prior art reference discloses, expressly or under principles of inherency, each and every element of a claimed invention.

The Final Office Action mis-states and oversimplified Applicant's position (Final Office Action, Page 3, Paragraph 4), suggesting that Applicant argues that Ratcliff "does not contain slave processes and sending to each slave process data that indicates which part of a set of information should be performed by that slave process". However, this remark is not an accurate paraphrasing of Applicant's position, nor does this remark track with the language of Claim 1.

As shown above, Claim 1 reads, inter alia, "sharing *the* set of information with *each* slave process of said plurality of slave processes" (emphasis added). In this limitation, "the information" refers to "a set of information about how to execute the database statement" recited earlier in Claim 1.

Note that the italicized 'the' conveys that the *same* set of information (about how to execute the database statement) is shared with *each* slave process. Meanwhile, the whole point of the Ratcliff invention is to *not* share the same information with each processor 105-130, but instead sub-divide the overall task so that each processor 105-130 knows about the *portion* of the task that the processor is supposed to perform (see at least FIG. 3, column 7 lines 44-58). Thus, the rejection of Claim 1 is flawed in that it appears to be based on a misunderstanding and misinterpretation of what Applicant means by "set of information".

This misunderstanding within the rejection of Claim 1 is reinforced by the fact that when Ratcliff engages in the process of parallelization, the distributing server 210 distributes algorithms and/or algorithm portions to various respective processors 105-130 (Ratcliff, col. 7, lines 50-54). However, those processors are only sent information about their specific algorithm that they are responsible for. Sometimes more than one processor will receive the same algorithm (Ratcliff, col. 7, lines 55-58). Nonetheless, in all instances within Ratcliff, the processors 105-130 will only execute the algorithm that it is sent, and will not know anything about any other algorithms being sent to any other processors. Thus, Ratcliff's algorithm cannot correspond to the claimed "set of information". If Applicant were claiming "sharing separate sets of information with each slave process", that would be different. However, that is not what Applicant is claiming, so that the rejection is based on a flawed premise and should therefore be withdrawn.

Additionally, Claim 1 also recites sending to each slave process data that indicates which part of a set of information should be performed by that slave process. It would be impossible for Ratcliff to disclose this, as Ratcliff's distributing server does not give any information about any other processes to a processor. Thus, there would be no need for Ratcliff to indicate which part of a set of information a processor should execute. The Ratcliff processor would not be in a position to make such a choice. This deficit was never addressed in any Office Action.

The rejection of Claim 1 also has an error in misinterpreting a portion of part (b) therein. Claim 1, which recites among other things, sharing the set of information with (as emphasized) *each* slave process, including (a) information about a task to be performed by said

slave process, and (b) information about one or more tasks, to be performed by processes *other than* the slave process, to execute the database statement (emphasis added).

Conversely, Ratcliff's distributing server does not give any information to a processor 105-130 other than the algorithm that specific processor is responsible for executing. Within Ratcliff, a policy of giving processors only what they themselves need to know makes sense, as a key purpose of Ratcliff is to take a large expensive complex task and divide it into smaller less complex more digestible tasks (see e.g. col. 9, lines 13-53).

However, Ratcliff's policy would not make sense if adapted to fit to the language of Claim 1, as the claimed "set of information" made available to each slave is not limited to the portion of the plan that is specific to the role of that particular slave in the execution of the plan. Because each slave is exposed to the "set of information", the slave can make intelligent decisions with respect to how to execute its portion of the execution plan. (Applicant's Specification, paragraph 0029). This reinforces the idea that the arguments contained in the Office Action suggest an incomplete or inaccurate understanding of what Applicant means by the claimed "set of information".

For at least the above reasons, Claims 1 and 15, as well as all claims dependent therefrom, are patentable over Ratcliff under 35 U.S.C. § 102(e). The rejections of Claims 1 and 15 are invalid and should be withdrawn.

By virtue of their dependence from Claim 1, Claims 2-14, 16-25 and 27-29 inherit the features that are distinguished from Ratcliff. Consequently, Claims 2-14, 16-25 and 27-29 are also patentable over Ratcliff, taken either individually or in combination, under 35 U.S.C. § 102(e).

B. <u>The Features of Claims 2 and 16 Are Not Disclosed, Taught, or Suggested by Ratcliff</u>

Claim 2 recites, inter alia, sharing an execution plan for the database statement. Ratcliff does not disclose an execution plan at all, much less sharing an execution plan for a database statement. The Hallmark reference, which was not applied to Claim 2 or 16 by the Examiner but belonging to the same Assignee as the instant application, discloses a type of execution plan, as well as slave processes, but in a context that can not correspond to Claims 2 or 1. Neither Ratcliff nor Hallmark disclose generating a duplicate execution plan. "The test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure" but rather "what the combined teachings of the references would have suggested to those of ordinary skill in the art." See In re Keller, 642 F.2d 413, 208 USPQ 871 (CCPA 1981).

Furthermore, Hallmark's sole original execution plan could not be duplicated without breaking the purpose of the invention disclosed therein. If a proposed modification would render a prior art invention being modified so as to be unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. In re Gordon, 7833 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984), *see also* MPEP 2143.01 V.

For at least the above reasons, Claims 2 and 16, as well as all claims dependent therefrom, are patentable over Ratcliff under 35 U.S.C. § 102(e).

C. The Features of Claims 6 and 20 Are Not Disclosed, Taught, or Suggested by the combination of Ratcliff and/or Hallmark

Claim 6 recites, inter alia, generating an execution plan for the database statement; constructing a shared cursor for the database statement, wherein the shared cursor provides

access to the execution plan. Hallmark discloses a cursor but it is not a shared cursor, and does not provide access to any execution plan.

The portion of Hallmark (column 26, lines 30-40; element 720, FIG. 7B) relied upon by the Final Office Action (Page 11, Paragraph 2) to correspond with this feature is inapplicable first because these portions discuss a non-shared cursor, and second because that non-shared cursor is not in any way connected to any execution plan. To establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. In re Royka, 490 F.2d 981, 180 USPQ 580 (CCPA, 1974). These deficiencies were never addressed within any Office Action. In rejecting a claim, the grounds of that rejection must be fully and clearly stated. See MPEP § 707.07(d).

Further, the Final Office Action cites Hallmark's Data Flow Operators (DFO) as corresponding to the claimed slaves (Final Office Action, Page), but earlier cited Ratcliff's Distributed Objects Environment (DOE) as doing so (Final Office Action, Page). Thus, the Office Action is inconsistent and contradictory regarding which portions of the cited prior art correspond to the claimed slaves.

For at least the above reasons, Claims 6 and 20, as well as all claims dependent therefrom, are patentable over the combination of Ratcliff and Hallmark under 35 U.S.C. § 103. The rejections of Claims 6 and 20 are invalid and should be withdrawn.

D. The Features of Claims 7 and 21 Are Not Disclosed, Taught, or Suggested by the combination of Ratcliff and/or Hallmark

Claim 7 recites, inter alia, allowing two or more of said slave processes to access a shared instance of the shared cursor. The portion of Hallmark (column 26, lines 57-67; element

742, FIG. 7C) relied upon by the Office Action (Page 11, Paragraph 3) to correspond with this feature is inapplicable because Hallmark does not disclose multiple instances of the same cursor, nor the DFOs nor slave processes sharing an instance of the cursor, as claimed. As stated in Royka, to establish prima facie obviousness of a claimed invention, **all** the claim limitations must be taught or suggested by the prior art (emphasis added). As shown above, several of the limitations of Claim 7 have not been addressed within any Office Action.

For at least the above reasons, Claims 7 and 21 are patentable over the combination of Ratcliff and Hallmark under 35 U.S.C. § 103. The rejections of Claims 7 and 21 are invalid and should be withdrawn.

E. The Features of Claims 11 and 25 Are Not Disclosed, Taught, or Suggested by the combination of Ratcliff and/or Hallmark

Claim 11 recites, inter alia, a granule iterator row source that encapsulates a horizontal partitioning of a base object upon which the database statement operates. The portion of Hallmark (column 5, lines 26-36) relied upon by the Office Action (Page 13, Paragraph 4) to correspond with this feature is incomplete. Although Hallmark discloses an iterator, there is still a substantial gap between an iterator and a granule iterator row source that encapsulates a horizontal partitioning of a base object upon which a database statement operates. Specifically, Hallmark does not disclose whatsoever a horizontal partition, and thus cannot disclose a granule iterator row source that encapsulates a horizontal partitioning of a base object upon which a database statement operates, as claimed. Again, several of the limitations of Claim 11 have not been addressed within any Office Action.

For at least the above reasons, Claims 11 and 25 are patentable over the combination of Ratcliff and Hallmark under 35 U.S.C. § 103. The rejections of Claims 11 and 25 are invalid and should be withdrawn.

CONCLUSION AND PRAYER FOR RELIEF

Based on the foregoing, it is respectfully submitted that the rejections of Claims 1-25 and 27-29 lack the requisite factual and legal bases. Appellants respectfully request that the Honorable Board reverse the rejections of Claims 1-25 and 27-29.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

/ChristopherMTanner#41518/
Chris Tanner

Date: January 27, 2008

2055 Gateway Place, Suite 550
San Jose, California 95110-1083
Tel: (408) 414-1238
Fax: (408) 414-1076

VIII. CLAIMS APPENDIX

1. A method for processing a database statement within a database server, the method comprising the steps of:

receiving at the database server the database statement;

determining that at least one operation required by the database statement can be parallelized;

within the database server, generating a set of information about how to execute the database statement;

causing a plurality of slave processes to perform said at least one operation by

sharing the set of information with each slave process of said plurality of slave processes, wherein the set of information shared with each slave process includes

(a) information about a task to be performed by said slave process, and

(b) information about one or more tasks, to be performed by processes other than the slave process, to execute the database statement; and

sending to each slave process of said plurality of slave processes data that indicates which part of the set of information shared with the slave process represents the part of the at least one operation that should be performed by the slave process.

2. The method of Claim 1 wherein:

the step of sharing the set of information includes sharing an execution plan for the database statement; and

sharing the execution plan with a particular slave process of the plurality of slave processes is performed by:

providing an original statement of the database statement to a node on which the particular slave process resides, wherein the original statement is the form of the database statement in which the database statement was received by the database server;

at said node, generating an equivalent execution plan based on the original statement; and

the particular slave process accessing the equivalent execution plan.

3. The method of Claim 2 wherein:

further comprising the step of providing to the node additional information that includes at least one of (a) values associated with session parameters of a database session in which the database statement was received, and (b) values associated with optimizer parameters that were used by an optimizer to generate a plan for the database statement in a node other than said node; and

the step of generating an equivalent execution plan is performed based, at least in part, on the additional information.

4. The method of Claim 1 wherein:

the step of generating a set of information includes generating an execution plan for the database statement, wherein the set of information includes the execution plan; and

the step of sending to each slave process of said plurality of slave processes data that indicates which part of the at least one operation should be performed by the slave

process includes sending to each slave process data that indicates a specific portion of the execution plan that is to be performed by the slave process.

5. The method of Claim 4 wherein:

the step of sending to each slave process data that indicates a specific portion of the execution plan that is to be performed by the slave process includes sending to a particular slave process data that indicates a particular portion of the execution plan that is to be performed by the particular process; and

the method further includes the step of the particular slave process determining how to execute the particular portion based, at least in part, on characteristics of the execution plan other than the particular portion of the plan that is to be executed by the particular slave process.

6. The method of Claim 1 wherein:

the step of generating a set of information includes

generating an execution plan for the database statement;

constructing a shared cursor for the database statement, wherein the shared cursor provides access to the execution plan; and

the step of sharing access includes providing each slave process of said plurality of slave processes access to the shared cursor.

7. The method of Claim 6 wherein the step of providing each slave process of said plurality of slave processes access to the shared cursor includes allowing two or more of said slave processes to access a shared instance of the shared cursor.

8. The method of Claim 6 wherein the step of providing each slave process of said plurality of slave processes access to the shared cursor includes allowing one of the slave processes to access a first instance of the shared cursor, and allowing another one of the slave processes to access a second instance of the shared cursor.

9. The method of Claim 8 wherein:

the one slave process resides on a first node;

the other slave process resides on a second node; and

the first node is a different node than said second node.

10. The method of Claim 9 wherein:

a first plurality of slave processes on said first node share access to said first instance of said shared cursor; and

a second plurality of slave processes on said second node share access to said second instance of said shared cursor.

11. The method of Claim 1 wherein:

the step of generating a set of information includes generating an execution plan for the database statement, wherein the set of information includes the execution plan; and

the method further comprises the step of inserting into the execution plan a granule iterator row source that encapsulates a horizontal partitioning of a base object upon which the database statement operates.

12. The method of Claim 1 wherein:

the step of generating a set of information includes generating an execution plan for the database statement, wherein the set of information includes the execution plan; and

the method further comprises the step of inserting into the execution plan at least one distribution row source that specifies how data is to be redistributed between one of

a first slave set and a query coordinator; and

a first slave set and a second slave set.

13. The method of Claim 12 wherein the step of inserting into the execution plan at least one distribution row source includes:

inserting into the execution plan at least one sender-side distribution row source that indicates how sending processes are to distribute data that the sending processes produce; and

inserting into the execution plan at least one receiver-side distribution row source that indicates how receiving processes are to obtain data that the receiving processes are to consume.

14. The method of Claim 1 wherein:

the step of generating a set of information includes generating an execution plan for the database statement, wherein the set of information includes the execution plan; and

the method further comprises the step of inserting into the execution plan a parallelizer row source that encapsulates the scheduling of tasks that slave processes are to perform.

15. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 1.

16. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 2.

17. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 3.

18. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 4.

19. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 5.

20. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 6.

21. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 7.

22. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 8.

23. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 9.

24. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 10.

25. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 11.

26. (cancelled)

27. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 12.

28. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 13.

29. A computer-readable storage medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in Claim 14.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.